

**Rapport personnel de projet :**

**5ème séance 31/01/2022**

**A la maison**

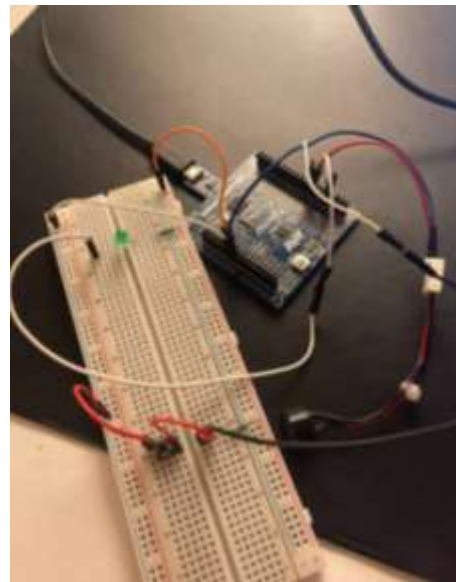
J'ai fait le code et le montage pour le buzzer qui est censé s'activer aux bons horaires. Ce code permet d'arrêter la Led et l'alarme lorsqu'on appuie le bouton poussoir. Il ne manque plus que l'associer au code de l'horloge.

```
const int led = 3;
const int bouton = 7;
int val=0;

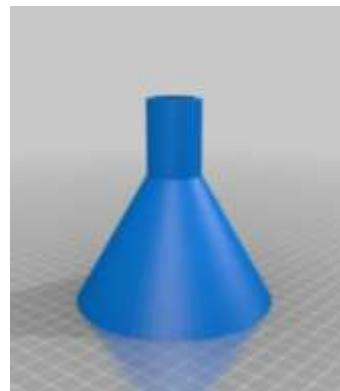
#define buz 6
const int buzzer = 4;

void setup() {
  Serial.begin(9600);
  pinMode(buzzer, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(bouton, INPUT);
  digitalWrite(led, LOW);
  delay(1000);
  tone(buzzer, 500);
}

void loop() {
  val = digitalRead(bouton);
  Serial.println(val);
  if (val==0) {digitalWrite(led, HIGH);}
  if (val==0) {noTone(buzzer);}
```



J'ai également fait les croquis pour la découpe laser et l'impression 3D pour créer la structure au FabLab.



Enfin, j'ai commencé les recherches sur le module Bluetooth et donc refamiliarisé sur ce que l'on avait vu en classe (code, application...).

## En cours

Durant la première partie de l'heure je suis partie au FabLab pour créer la boîte. Après la découpe laser, j'ai récupéré les plaques en bois mais celles-ci avaient un défaut dans les encoches donc je les ai poncées, puis nettoyées, assemblées et collées.



Une fois cette partie qui a pris plus de temps que prévu il me restait que 45min pour commencer réellement le Bluetooth. J'ai donc fait le montage et le code.

Problème : le module ne me renvoyait pas « OK » lorsque je tapais « Bonjour – Prêt pour les commandes AT » comme on avait fait dans le cours.

Code d'essai 1

```
Bluetooth
#include <SoftwareSerial.h>
#define RX 10
#define TX 11

SoftwareSerial BlueT(RX, TX);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  delay(500);
  Serial.println("Bye, good for the commands AT");
  BlueT.begin(9600);
  delay(500);
}

void loop() {
  // put your main code here, to run repeatedly:
  while (BlueT.available()) {
    Serial.print(char(BlueT.read()));
  }
  while (Serial.available()) {
    BlueT.write(char(Serial.read()));
  }
}
```

Code d'essai 2

```
#include <SoftwareSerial.h>
#define RXPin 11 // Receive 11 au bout que RX, à connecter aux TX du HC-05
#define TXPin 10 // Transmit 10 au bout que TX, à connecter aux RX du HC-05
SoftwareSerial mySerial(RXPin, TXPin);

void setup()
{
  // define pin modes for rx, tx pins:
  pinMode(RXPin, INPUT);
  pinMode(TXPin, OUTPUT);
  mySerial.begin(57600);
  Serial.begin(57600);
}

void loop()
{
  int i = 0;
  char someChar[32] = {0};
  // when characters arrive over the serial port...
  if (Serial.available()) {
    do {
      someChar[i++] = Serial.read();
    } while (1);
    Serial.println(someChar);
    mySerial.print(someChar);
    Serial.print(someChar);
  }
  while (mySerial.available()) {
    Serial.print(char(mySerial.read()));
  }
}
```

J'ai essayé un autre code mais je ne captais pas le module sur l'application. J'ai donc, jusqu'à la fin de l'heure, essayé au moins de refaire ce que l'on avait fait en classe en vain sans comprendre pourquoi.

**Objectif pour la prochaine séance :**

- Si Lisa a fini le montage parent, j'associerai mes codes de la led+alarme+bouton pour faire le fonctionnement général du distributeur
- Solutionner le Bluetooth et le finir si possible